More about proofs

François Thiré

February 3, 2019

Review from last time

- `by []`
- `move`
- `case`
- `apply`
- `elim`

Figure 1: Tactics

- `by ...`
- `:`
- `=>`
- `;`
- `[...|...|...]`

Figure 2: Tacticals

Construct new proofs

Ssreflect vs Mathcomp

Construct new proofs

Ssreflect vs Mathcomp

Should you use the standard library?

- Yes: no need to reinvent the wheel!
- No: it is a great exercise to reproof the lemmas you need

Construct new proofs

Ssreflect vs Mathcomp

Should you use the standard library?

- Yes: no need to reinvent the wheel!
- No: it is a great exercise to reproof the lemmas you need

My advice: If you to prove it in Coq, don't prove it again!

Searching definitions and theorems

Use the command `Search`!

It takes an arbitrary number of arguments.

Searching definitions and theorems

Use the command `Search`!

It takes an arbitrary number of arguments.

- Argument can be a string: `Search "add"`
- Argument can be a pattern: `Search (_ + _)`

- The first argument if it is a pattern matches only in the conclusion of the lemma.
- Arguments can be combined: `Search "K" (_ + _)`

Search engine is not modulo computation!

```
Search _ ((_+_) = (_+_)).
About addnC.
Eval hnf in commutative addn.
```

Give proper names to your lemmas!

Conventions used in SSreflect:

- `fee_fie_foe` says something about `(fee .. (fie ..(foe ..) ..) ..)`
- Standard properties (Commutativity, Associativity, Kancellation lemmas) have a dedicated suffix.

Read section 2.5 of the mathcomp book!

Project

Projet is online!

- A lot of different and independent exercises
- Hardness of the question is indicated by $\star$
- For three stars questions and beyond: Think on paper before!
- Further lectures will introduce the tools you might need!

Evaluation:

- Trials and errors are welcome!
- Constructive pidgeon hole principle's proof should give you the average.

Have fun!